Practical assignment 5 (R05) (Building your first deep neural network on page 130 of the Grokking Deep Learning textbook)

Consider the following code snippet which implements the backpropagation learning algorithm:

```
import numpy as np
np.random.seed(1)

def relu(x):
    return (x > 0) * x

def relu2deriv(output):
    return output>0

streetlights = np.array([[1, 0, 1], [0, 1, 1], [0, 0, 1], [1, 1, 1]])
walk_vs_stop = np.array([[1, 1, 0, 0]]).T

alpha = 0.2
hidden_size = 4

weights_0_1 = 2*np.random.random((3,hidden_size)) - 1
weights_1_2 = 2*np.random.random((hidden_size,1)) - 1

for iteration in range(60):

    layer_2_error = 0

    for i in range(len(streetlights)):

        layer_0 = streetlights[i:i+1]
        layer_1 = relu(np.dot(layer_0,weights_0_1))
        layer_2 = np.dot(layer_1,weights_1_2)
        layer_2_error += np.sum((layer_2 - walk_vs_stop[i:i+1]) ** 2)
        layer_2_delta = (layer_2 - walk_vs_stop[i:i+1])
        layer_1_delta=layer_2_delta.dot(weights_1_2.T)*relu2deriv(layer_1)

        weights_1_2 -= alpha * layer_1.T.dot(layer_2_delta)
        weights_0_1 -= alpha * layer_0.T.dot(layer_1_delta)

    if(iteration % 10 == 9):
        print("Error:" + str(layer_2_error))
```

Please modify the code mentioned above in the following manner:

1) Implement the following additional activation functions: Hyperbolic tangent, Linear, and Sigmoid.
2) Implement the derivatives of the three activation functions in Step 1.
3) By not changing the hyperparameters in the code snippet above, determine which of the four activation functions (Hyperbolic tangent, Linear, ReLu and Sigmoid) gives the smallest error.

Please write a report of no more than 1000 words on neural network activations functions in general. The report do not have to reference to your Python solution for this assignment specifically. Cite reputable sources to motivate your arguments. Submit your Python (.py) text file and Word document as two separate assignments on eFundi.